UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/866,375	05/25/2001	Ruiguo Yang	2006579-0155	4422

69665 7590 09/24/2008

CHOATE, HALL & STEWART / CITRIX SYSTEMS, INC.
TWO INTERNATIONAL PLACE
BOSTON, MA 02110

EXAMINER	
NGUYEN, DUSTIN	
ART UNIT	PAPER NUMBER
2154	
MAIL DATE	DELIVERY MODE
09/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte RUIGUO YANG and HENRY COLLINS

Appeal 2008-2771
Application 09/866,375
Technology Center 2100

Oral Hearing Held: August 13, 2008

Before JEAN R. HOMERE, ST. JOHN COURTENAY III, and STEPHEN C. SIU, *Administrative Patent Judges.*

ON BEHALF OF THE APPELLANT:

JOHN D. LANZA, ESQ.
Choate Hall & Stewart
Two International Place
Boston, Massachussetts 02110

The above-entitled matter came on for hearing on Wednesday, August 13, 2008, commencing at 10:57 a.m., at the U.S. Patent and Trademark Office, 600 Dulany Street, 9th Floor, Alexandria, Virginia, before Jennifer M. O'Connor, Notary Public.

THE CLERK: Calendar number 20, Mr. Lanza.

MR. LANZA: Good morning.

JUDGE HOMERE: Good morning, Counselor. You have 20 minutes and feel free to begin whenever you're ready.

MR. LANZA: Thank you. Thank you. My name is John Lanza, PTO registration number 40060, and I represent Citrix Systems in appeal 2008-2771. This appeal comes to this board on the rejection of all pending claims as obvious in view of three prior art references. The claims are currently pending in serial number 09866375.

Citrix Systems asks this board to reverse the examiner's rejection of all the pending claims as obvious and remand this application back to the examiner with instructions to allow it over the prior art of record.

JUDGE COURTENAY: I do have to say, we have the power to affirm or reverse, or affirm in part, but the examining corps makes the ultimate decision as to what is allowed. We don't allow any cases here at the Board.

MR. LANZA: Understood. Thank you.

JUDGE COURTENAY: Could you go to the preamble issue to begin; that would be helpful for us? I understand that the examiner, for the first time in the prosecution history, he's saying that he's not giving certain limitations in the preamble patentable weight?

MR. LANZA: That is what it appears to be. In his examiner's reply, the examiner stated twice -- sorry, the examiner's answer, the examiner stated twice that limitations from the specification are not read into the

claims and that the preamble had not been given patentable weight because the recitation that we were relying on occurs in the preamble

If we can turn to the claims briefly. And I thank you for bringing this issue up, because I admit that I was struggling with -- I've got my argument, but now I have this new issue that I have to deal with, so I appreciate having the chance to just deal with --

JUDGE COURTENAY: Well, we do commend you for the proper use of a Reply Brief. The examiner has raised a new issue in the Answer for the first time and you have properly responded to it in the Reply Brief.

MR. LANZA: Thank you. It's our position that the claims, as they're currently pending, the preamble is limiting for two reasons. One, the claim elements rely on the preamble for antecedent basis, which I believe is a well-settled basis for viewing the preamble as a claim limitation. There's also a doctrine that we refer to in our reply brief that states that if the applicant relies on the preamble in his or her arguments for patentability, that that is -- that is then an indication that the preamble should be considered a claim limitation. I believe that's the Catalina Marketing case that we pointed to in our Reply Brief.

And if we -- if we look at claim 1, claim 12 is similar. But claim 1 recites a method for controlling by a server the formation of an off-screen surface at a client, and then I'll allay that and we'll look at the first claim element instructing the client to select a first memory region. There's no antecedent basis for client other than in the preamble, and so we have always viewed this claim -- we continue to view this claim as a client server

implementation where the server is instructing the client to perform specific tasks.

The server is instructing the client to create or form an off-screen surface buffer in its memory. It then transmits in the second step an indicia of some graphical data to the client. Graphical data is referred to in the specification at paragraph five as, for example, bit-mapped graphical data, encoded bitmaps, glyphs and line data.

The reason that an indicia of the graphical data is sent to the client is that this entire application is specifically directed to techniques to minimize the bandwidth necessary to communicate between a client and a server. The server then instructs the client to copy the graphical data associated with the indicia to a particular location in that first memory region that it has been instructed to set up.

So a bitmap is sent down. It's cached at the client. There's a cache tag of some sort associated with that bitmap or that line. That cache tag has been sent back to the client. The client is instructed by the server to collect that bitmap and write it into its off-screen buffer.

That is crucial to understanding the arguments that we've made throughout. And in fact, up until the examiner's answer, we had understood the examiner to be saying that all of the prior art references -- or I should say not all -- the two major prior art references, which are *CLAPP* and *Hanko*, both taught a server instructing a client to form an off-screen buffer and to store data in that off-screen buffer.

I can go through why neither *CLAPP* nor *Hanko* teach that, teach a server instructing a client to do that. In the answer -- this appears for the

first time -- to be in the examiner's Answer for the first time. The examiner appears to have realized that *Hanko* and *CLAPP* do not teach that and instead relies on a broader reading of the claim, which tries to read out this server requirement in order to make *Hanko* and *CLAPP* apply.

JUDGE HOMERE: Let's turn to what the prior art teaches. From what I understand from the record, the prior art teaches you have two computers that are communicating, right, they're sharing information. So one computer accesses or opens up an application and allocates opening that application, allocates an off-screen to a buffer, right, an off-screen surface to a buffer, and then sends communication to the other computer and the other computer operates in the same fashion and it pretty much transmits whatever image that they're looking at to the other computer?

MR. LANZA: I think that's right, Your Honor. I believe we're talking about CLAPP right now, which is a videoconferencing system which performs as you describe. There are -- in *CLAPP*, it's a symmetric system in that there is a local host and a remote host connected by a communication channel, which is number 82 in *CLAPP*, and the local host is described as creating a local off-screen buffer into which it writes local pixel data.

The local host and the remote host are able to share their off-screen buffers and then the remote host will copy into its display buffer from its own off-screen buffer. So what is missing from *CLAPP* is that in *CLAPP*, the local host never instructs the remote host to one, create an off-screen buffer, because it already exists. The remote host has to have it because it's a symmetrical system.

JUDGE HOMERE: There's nothing in the claim that says that requires the server to instruct the client to create an off-screen buffer. It instructs the client to select -- off-screen buffer. And so wouldn't it be reasonable for one to construe the teaching of *CLAPP* as saying that once the client, the first client transmits the image to the other client, a buffer, an off-screen buffer is allocated subsequently to receiving that message from the other client?

MR. LANZA: If you were to interpret *CLAPP* that way -- I don't believe that *CLAPP* does that. I believe that *CLAPP* always has that buffer allocated at startup time. For the sake of argument, if I were to interpret *CLAPP* to disclose that when the local host sends data to the remote host, that upon receipt of that data the remote host creates its own buffer, the local host is not instructing the client that -- it's not instructing the remote host to create that buffer.

JUDGE HOMERE: Why not?

MR. LANZA: It's simply -- it's simply sending data. Your point though, I thought, was a good one and an accurate one and I appreciate you raising it, which is the claim language is actually instructing the client to select a first memory region for allocation to the off-screen buffer. Again, that -- one could argue whether selection or creation are different. They are, because in the -- in our specification, we talk about having two areas in the memory region and being able to select between them for the data. But the transfer of data from CLAPP -- in *CLAPP* from the remote host to the local host, is merely just that; it's a transfer of data.

JUDGE HOMERE: Yeah, but the allocation of a buffer, if you will, in the remote host is subsequent to receiving that data. Therefore --

JUDGE COURTENAY: Well, it actually says in column 12 of *CLAPP*, lines three and four, that the communication may proceed subsequently to or concurrently with the processing step. So the transmission of the data can happen concurrently or subsequent to what happens at the local host.

I've been reviewing *CLAPP*, Figure 12, and particularly columns 11 and 12, and in Figure 12, in block 652, we have a disclosure that at the remote site there's this copying of local pixel data to the remote off-screen window buffer. But we don't have a real teaching that tells us when that buffer's allocated, more specifically, how it's allocated, other than as you raised the point that we have symmetric systems here.

We know they're symmetric because in column 12, lines seven through eight, it discloses -- *CLAPP* discloses a remote host computer system 264 preferably operates visual conferencing application software substantially similar to that operating on a local host computer system 244.

JUDGE HOMERE: My question still stands though, in the sense that, well, is that a possible interpretation of that teaching? Wouldn't it be reasonable for one of ordinary skill in the art to say that, well, I have this local client and then upon opening an application, it allocates a buffer, sends -- transmit data to a remote -- a remote client and then the remote client allocates -- subsequently to receiving that data, allocates an off-screen buffer in order to establish the communication?

MR. LANZA: I respectfully submit, Your Honor, that it's not a reasonable interpretation and this is why. In the system of CLAPP, both hosts have to be operating the same software, so the software has to behave the same whether it's a local host or a remote host. That off-screen buffer that is created on one of the hosts, I don't care which one, is created when the videoconferencing system is set up, because the off-screen buffer is used to move data in and out of the windows on the computer that is attached to the videoconferencing host. So that off-screen buffer is always going to be there when the system is operating.

JUDGE HOMERE: So you are assuming -- you are assuming that both of them -- both the local and the remote clients launched that same application before they start communicating, therefore, because -- from the teaching of *CLAPP*, upon launching that application, that buffer is created, right?

JUDGE COURTENAY: Actually, that's -- he's pointing to column 11 and we do have information about how the local host computer system allocates the buffer at lines 32 through approximately 36 of column 11. At line 32, it discloses -- the *CLAPP* reference discloses the user at step 628 then selects a local active application window 602 from the menu 600 for sharing with a remote conferencing site.

JUDGE HOMERE: That is the local --

JUDGE COURTENAY: The local host computer system 244 at step 630 preferably allocates an appropriate amount of system memory to accommodate a local on-screen window buffer 604. So we know how the local host system allocates its memory and we also have information that

these two systems, the remote host and the local, are symmetric. They operate the same way. We don't appear to have information as to the timing of how this happens at the remote site beyond what we have with respect to the local host.

JUDGE HOMERE: And my question, as a way to fill the gap, as far as the information that's missing from this reference, wouldn't it be reasonable to say that while you start with the local host first, you launch that application, the buffer's allocated, you transmit data over to the remote host, and the remote host allocates the buffer?

MR. LANZA: If it is reasonable to assume that, that is not a teaching of this specification. Now you're requiring the specification to act as an obviousness reference, because this specification does not teach, as required by the law, that command to set up the off-screen buffer at the remote host. An examiner is relying on *CLAPP* to teach that one of the -- one of the hosts instructs the other host to create the local buffer, which it doesn't do; it doesn't teach that.

JUDGE HOMERE: What I'm saying is, if the local buffer is created, if the remote buffer is created subsequently to receiving the data from the local host, wouldn't that be construed as -- the creation of the remote buffer, the buffer at the remote host becomes construed as being created based on an instruction received from the local host?

MR. LANZA: Yes, Your Honor, but could also be construed as being created, because the remote host has decided that it wants to create an off-screen buffer. It may receive an indication from the local host that data is to be shared, and it elects not to set up an off-screen buffer. It may elect to just

direct that data directly into the on-screen buffer because of various conditions. Usually off-screen buffers are used when windows are clipped, and you don't want to show some of the data that's actually in the window, so you write that to an off-screen buffer.

There's no -- there's no teaching at all on how the local host is going to decide whether or not to set up an off-screen buffer and there's certainly --

JUDGE HOMERE: The remote host, you mean?

MR. LANZA: I may have misspoke, I'm getting confused with local host or remote host at this point. They're both hosts. But whichever -- whichever system is getting the data from the other system, there's no -- there's no teaching at all on how or whether that system decides to set up an off-screen buffer. We just know that there is -- there is later some discussion about the remote host could have an off-screen buffer there.

JUDGE COURTENAY: The *CLAPP* reference doesn't appear to disclose exactly what triggers this allocation. If I look at Figure 12, step 628, the user selects this local active window application and then subsequent to that step, we have this memory allocation step 630. There's really no information that I can see from this reference that tells us what triggers this allocation.

MR. LANZA: I agree, and that's our position, that this does not teach one of the systems instructing the other to create or to select memory for an off-screen buffer. We don't know -- *CLAPP* doesn't teach how that gets selected.

JUDGE HOMERE: But you would agree though -- you would agree though if -- if the creation of the buffer at the remote host is done subsequent to receiving that image or that data from the remote -- the local host, that could be construed as an instruction that's creating the buffer at the remote host upon receiving an instruction from the local host?

(Pause)

JUDGE COURTENAY: Don't you argue in your Briefs that the mere receiving of data is not receiving an instruction?

MR. LANZA: Yes, and I'm trying to decide how to respond to Your Honor's question because it could be construed to be that, but I would -- I would argue that a command . . . , we've made a distinction in paragraph two between graphical data and other types of network traffic. And so an instruction is not graphical data. So an instruction is not data that might come from an off-screen buffer such as a bit-mapped -- bit-mapped graphical data, encoded bitmap, glyphs or line data.

It's something different that is an explicit indication to the other host, you need to set up an off-screen buffer because I'm about to throw -- I'm about to throw data at you and I want you to store it in there.

JUDGE HOMERE: Do you have anything else?

MR. LANZA: I do. The examiner, in his reply, has admitted that the other two elements of claim 1 are not in *CLAPP*. I think it is worth speaking briefly about *Hanko*, because the examiner refers to *Hanko*, which is a Sun Microsystems patent, that talks about efficient methods of displaying tiled graphical data at a remote computer.

It appears that from *Hanko*, *Hanko* teaches that while off-screen buffers are good, off-screen buffers aren't available at all devices and so this is a technique for efficiently replicating tile data in systems where one does not want to use or cannot use an on-screen buffer. We make these arguments in the Brief as well.

As you know from our Briefs, I don't believe *Hanko* should be considered as prior art, as analogous prior art, at all because one of ordinary skill in the art looking for ways to improve off-screen buffering techniques in a client server system would not look to a reference that in its beginning pages says, off-screen buffers are wonderful, but lots of computers don't have off-screen buffers so we're going to talk about techniques for doing this without using off-screen buffers.

Hanko also, therefore, does not teach instructing a client to select a first memory region to be used as an off-screen buffer, because it teaches away from off-screen buffers. It doesn't want you to use off-screen buffers. In fact, it assumes that you're going to receive -- that the client is going to receive that graphical data from the server and write it directly to the on-screen buffer and use some techniques to replicate that tiled data over and over and over again.

In fact, the techniques that he uses is that with respect to bit-mapped data. It sends some attributes of that data. It sends the size. It sends the repetition number, and then it uses -- the client uses that information to repeat the bit-mapped tile.

So in fact, the second step, which is transmitting indicia of a graphical data, the examiner's taken the position that indicia can be almost anything.

However, if you look at our specification at paragraph 71, we talk about what indicia is and it's about midway down through the paragraph. But it says that the server agent issues a command to the client agent to form the off-screen surface in the client's volatile memory 114. The command is then accompanied by an encoded representation of the off-screen surface if this is the first instance of the off-screen surface or by an index or a fuzzy key if the off-screen surface has been previously transmitted.

The indicia that is recited to in that step of claim 1, and a similar step in claim 12, is not attribute data about the graphical data that is coming across, but is really referring to cache indicia, such as a cache tag or a fuzzy key, which we talk about earlier in the specification. That point is reinforced by claim 2, which then further recites that step -- that claim 1 further comprises the step of specifying a plurality of attributes for the graphical data.

So *Hanko* does not teach or suggest instructing a client to select memory to form an off-screen surface. It also doesn't teach or suggest transmitting indicia of the graphical data to the client. I would submit that *Hanko* also doesn't teach or suggest instructing the client to copy the graphical data. At best, what *Hanko* teaches is that graphical data is sent to the client with instructions on how to copy.

Hanko describes or discloses that the client itself makes the decision to copy that bitmap glyph over and over and over again in its memory. So to sum up, because I believe I'm running out of time, it's our position that the preamble is limiting, that *CLAPP* does not teach or suggest instructing the

client to select a first memory region. At best, it teaches that there are two memory regions that are used for off-screen buffers.

The examiner admits that the other two steps of claim 1 and similar steps in claim 12 are not taught or suggested by *CLAPP*. *Hanko* does not teach or suggest these elements, for the reasons I've stated, and *Pierson*, which we haven't talked about much, but I think is a fairly minor reference here, also does not teach or suggest these. So any combination of these references is going to fail to teach or suggest each and every element of claims 1 and 12. So it's our view that the rejection of claims 1 through 20 is not obvious in view of these references, is improper and ought to be reversed.

JUDGE HOMERE: Anything else? All right, thank you very much.

MR. LANZA: Thank you.

(Whereupon, at 11:20 a.m., the proceedings were concluded.)